Amendments in the Claims:

Claims 5, 22 and 23 have been canceled. Claims 31-42 have been added.

Listing of the claims:

1. (Previously Presented) A method including:

dedicating a first portion of a resource exclusively to a first thread;

dedicating a second portion of the resource exclusively to a second thread; and

dynamically sharing a third portion of the resource between the first and second threads.

2. (Previously Presented) The method of claim 1 wherein the dynamic sharing of the third portion of the resource is performed according to resource demands of the respective first and second threads.

3. (Previously Presented) The method of claim 1 wherein the resource comprises a memory resource including first and second portions dedicated to the first and second threads respectively and a third portion shared between the first and second threads, the method including:

identifying a first location within the memory resource as a candidate location to receive an information item associated with the first thread;

determining whether the candidate location is within the first portion or the third portion of the memory resource dedicated to the first thread;

if the candidate location is within the first portion or the third portion of the

memory resource, then storing the information associated with the first thread at the candidate location; and

if the candidate location is within the second portion of the memory resource then identifying a further location as being the candidate location.

4. (Previously Presented) The method of claim 3 wherein the memory resource comprise a N way set associative memory and wherein the first portion comprises a first way dedicated to the first thread, the second portion comprises a second way dedicated to the second thread and the third portion comprises a third way shared between the first and second threads, wherein the identification of the first location as the candidate location comprises identifying a selected way within a selected set of the memory as a candidate way to receive the information item associated with the first thread.

5. (Canceled)

6. (Previously Presented) The method of claim 4 wherein the identification of the selected way within the selected set as the candidate way comprises identifying a way within the select set that was least recently used.

7. (Previously Presented) The method of claim 6 wherein the identification of the selected way within the selected set as the candidate way comprises identifying a way within the selected set that was second-least recently used.

8. (Previously Presented) The method of claim 6 including examining a Least Recently Used (LRU) history for the selected set to identify the way that was least recently used.

9. (Previously Presented) The method of claim 8 including examining a set of entries within the LRU history for the selected set, each entry within the set of entries indicating a respective way within the selected set, wherein the set of entries

is ordered in a sequence determined by least recent usage of a respective way and the selection of the candidate way comprises performing a sequential examination of the entries of the set of entries to locate a least recently used way that comprises either the first or the second way.

10. (Previously Presented) The method of claim 4 wherein memory comprises a trace cache memory, and wherein the information item associated with the first thread comprises a microinstruction of the first thread.

11. (Previously Presented) A resource comprising:

a first portion dedicated to utilization by a first thread executing within a multi-threaded processor;

a second portion dedicated to utilization by a second thread executing within the multi-threaded processor; and

a third portion shared by the first and second threads.

12. (Previously Presented) The resource of claim 11 wherein the resource comprises a memory including selection logic to identify a first location within the memory resource as a candidate location to receive an information item associated with the first thread, to determine whether the candidate location is within the first or third portions of the memory resource, then to store the information associated the first thread at the candidate location but, if candidate location is within the second portion of the memory resource, then to identify a further location as being the candidate location.

13. (Previously Presented) The resource of claim 12 comprising a N way set associative memory and within the first portion comprises a first way dedicated to the first thread, the second portion comprises a second way dedicated to the second thread and the third portion comprises a third way shared between the first and

second threads.

14. (Previously Presented) The resource of claim 12 wherein the selection logic identifies a selected way within a selected set of the memory as a candidate way to receive the information item associated with the first thread if the selected way comprises either the first or the third way.

15. (Previously Presented) The resource of claim 12 wherein the selection logic identifies a further way within the selected set of the memory as the candidate way to receive the information item associated with the first thread if the selected way comprises the second way.

16. (Previously Presented) The resource of claim 14 wherein the selection logic identifies the selected way within the selected set as the candidate way by identifying the selected way within the select set as a last recently used way within the selected set.

17. (Previously Presented) The resource of claim 15 wherein the selection logic identifies the further way within the selected set a candidate way by identifying the further way within the selected set as a second-least recently used way within the selected set.

18. (Previously Presented) The resource of claim 16 wherein the selection logic examines a Least Recently Used (LRU) history for the selected set to identify the way that was least recently used.

19. (Previously Presented) The resource of claim 18 wherein the selection logic examines a set of entries within the LRU history for the selected set, each entry within the set of entries indicating a respective way within the selected set, wherein the set of entries is ordered in a sequence determined by least recent usage of a respective way and the selection of the candidate way comprises performing a sequential examination of the entries of the set of entries to locate a least recently

used way that comprises either the first or the second way.

20. (Previously Presented) The resource of claim 18 wherein the memory comprising a trace cache memory, and wherein the information item associated with the first thread comprises a microinstruction of the first thread.

21. (Previously Presented) Selection logic including:

first means for identifying a first location within a memory resource, associated with a multi-threaded processor, as a candidate location to receive an information item associated with a first thread; and

second means for determining whether the candidate location is within a second portion of the memory resource dedicated to the second thread;

wherein, if the candidate location is within the second portion of the memory resource dedicated to second thread, the first means for identifying a further location within the memory resource as the candidate location, and wherein the memory resource comprises an N way set associative memory and wherein the first portion comprises a first way dedicated to the first thread, the second portion comprises a second way dedicated to the second thread and the third portion comprises a third way shared between the first and second threads, and wherein the first means is for identifying a selected way within a selected set of the memory as a candidate way to receive information not associated with the first way.

22. (Canceled)

23. (Canceled)

24. (Previously Presented) A method including:

defining a memory resource, associated with a multi-threaded processor, to include first and second portions dedicated to the first and second threads respectively and a third portion shared between the first and second threads;

for an information item associated with the first thread, examining a history of least recently used portions to identify either the first portion or the third portion as being a least recently used portion available to the first thread; and

storing the information item within the least recently used portion.

25. (Previously Presented) The method of claim 24 wherein, for the information item associated with the first thread, the second portion is excluded from the identification as the least recently used portion on account of being dedicated to the second thread.

26. (Previously Presented) The method of claim 24, wherein the memory resource comprises a N way set associative cache memory and wherein the first, second and third portions comprising respective first, second and third ways.

27. (Previously Presented) The method of claim 26 wherein the examination of the history of least recently used portions includes examining a least recently used history for a selected set of the set associative cache memory.

28. (Previously Presented) The method of claim 24 wherein the cache memory comprises a trace cache memory, and wherein the information item associated with the first thread comprises a microinstruction of the first thread.

29. (Previously Presented) A computer-readable medium storing a sequence of instructions that, when executed within a processor, causes the processor to perform the steps of:

dedicating a first portion of a resource exclusively to a first thread;

dedicating a second portion of the resource exclusively to a second thread; and

dynamically sharing a third portion of the resource between the first and second threads.

30. (Previously Presented) The computer readable medium of claim 29 wherein the dynamic sharing of the third portion of the resources is performed according to resource demands of the respective first and second threads.

31. (New) A processor comprising:

a cache divided into a plurality of partitions;

victim selection logic to,

allow replacement of a first information element from a first thread in a shared partition of the cache with a second information element from a second thread but to prevent replacement of a third information element from the first thread in a first partition of the cache with the second information element from the second thread; and

allow replacement of a fourth information element from the second thread in the shared partition of the cache with a fifth information element from the first thread but to prevent replacement of a sixth information element from the second thread in a second partition of the cache with the fifth information element from the first thread.

32. (New) The processor of claim 31 wherein the first partition is a first thread partition, the second partition is a second thread partition.

33. (New) The processor of claim 31, wherein the cache is a trace cache and the processor is capable of fine simultaneous multithreading.

34. (New) The apparatus of claim 31, wherein the cache is a multi-way set associative cache and each of the partitions includes one or more ways of the cache.

35. (New) The apparatus of claim 31, wherein the victim selection logic includes a least recently used scheme.

36. (New) A method comprising:

detecting misses in a cache of a processor;

performing victim selection responsive to each of the misses in a manner that partitions the capacity of the cache to make a first and second partition available for replacement respectively only by each of a first and second instruction threads while at the same time that defines a shared portion of the capacity of the cache that is available to both the first and second instruction threads.

37. (New) The method of claim 36, further comprising:
performing fine multithreading of the first and second threads in the processor.

38. (New) The method of claim 36, wherein the performing victim selection includes determining which of a plurality of entries in the cache is the least recently used.

39. (New) An apparatus comprising:
a processor including,

a cache having a storage capacity, and

victim selection logic including logic to partition the storage capacity of the cache with a bias toward having a dedicated portion for each of a first and second

instruction threads while at the same time having a shared portion accessible by both the first and second threads; and

a main memory coupled to the processor having stored therein the first and second instructions threads, which are of a multi-media type.

40. (New) The apparatus of claim 39, wherein the cache is a trace cache and the processor is capable of fine simultaneous multithreading.

41. (New) The apparatus of claim 39, wherein the cache is a multi-way set associative cache and each of the partitions includes one or more ways of the cache.

42. (New) The apparatus of claim 39, wherein the logic to partition is based on a least recently used scheme.